This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(12) UK Patent Application (19) GB (11) 2 334 680 (13) A

(43) Date of A Publication 01.09.1999

(21) Application No 9907163.1

(22) Date of Filing 29.03.1999

(71) Applicant(s)
Yehouda Harpaz
129 Corrie Road, CAMBRIDGE, CB1 3QQ,
United Kingdom

(72) Inventor(s)
Yehouda Harpaz

(74) Agent and/or Address for Service
Yehouda Harpaz
129 Corrie Road, CAMBRIDGE, CB1 3QQ,
United Kingdom

(51) INT CL^6
A63F 3/02

(52) UK CL (Edition Q)
A6H HLM

(56) Documents Cited
GB 2147817 A WO 97/26057 A1 JP 040367684 A
US 4279421 A US 4244635 A

(58) Field of Search
UK CL (Edition Q) A6H HLM
INT CL^6 A63F 3/00 3/02
Online: EPODOC, PAJ, WPI.

(54) Abstract Title
Electronic game board

(57) An electronic game board comprises an array of grid points and a controller. Each grid point is sensitive to touch and has a display associated with it. When a grid point is touched, the controller is notified and it displays the state of the game either by appropriate lights e.g. LEDs or on a LCD. The controller can be programmed so that various games can be played. The game board is particularly designed for playing games of the type where a stone is put on a board, the touching of a grid point being equivalent to the placing of a stone. Such games include GO, Othello and the "Game of Life". The board may also have a timer, randomisation, lights 5 to indicate whose turn it is, a control area 10, 11, 12 to allow a user to set which game is played and other parameters, and a score indicator. The board may also be programmed with puzzles and other exercises.

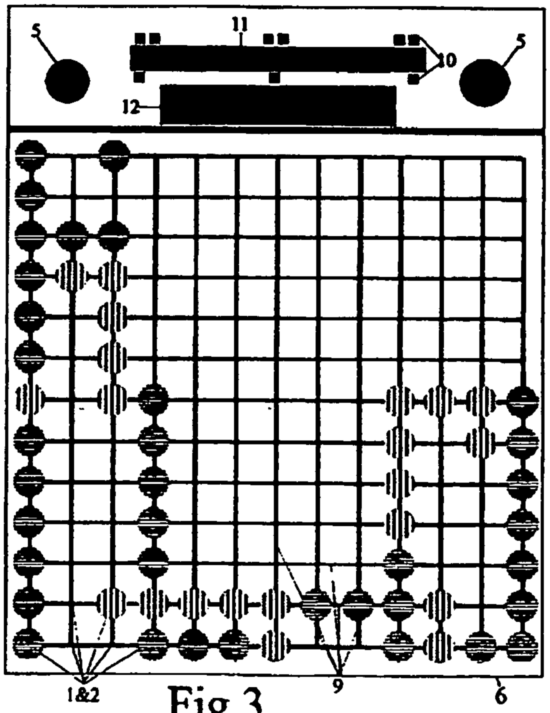


Fig 3

GB 2 334 680 A

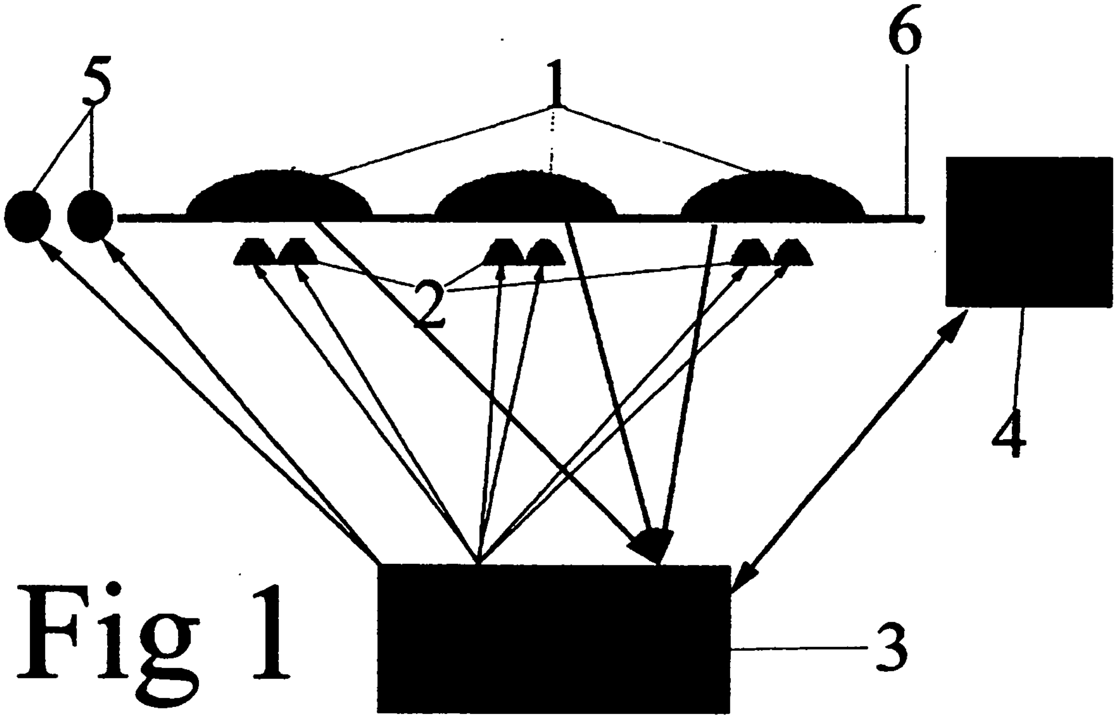
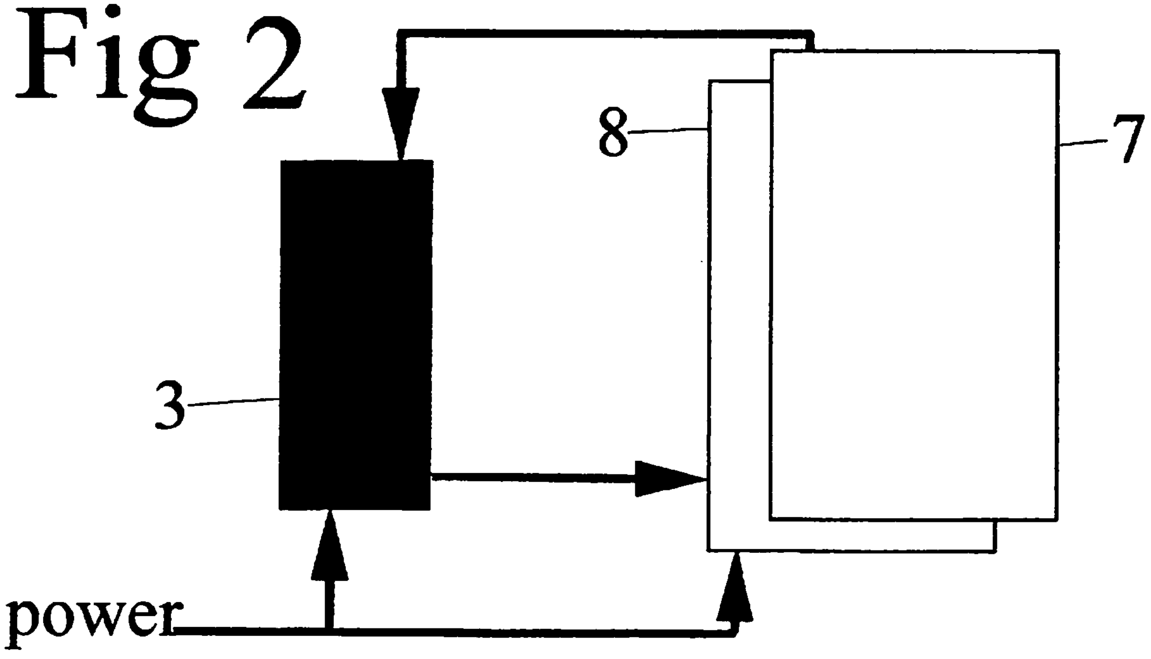


Fig 1

Fig 2



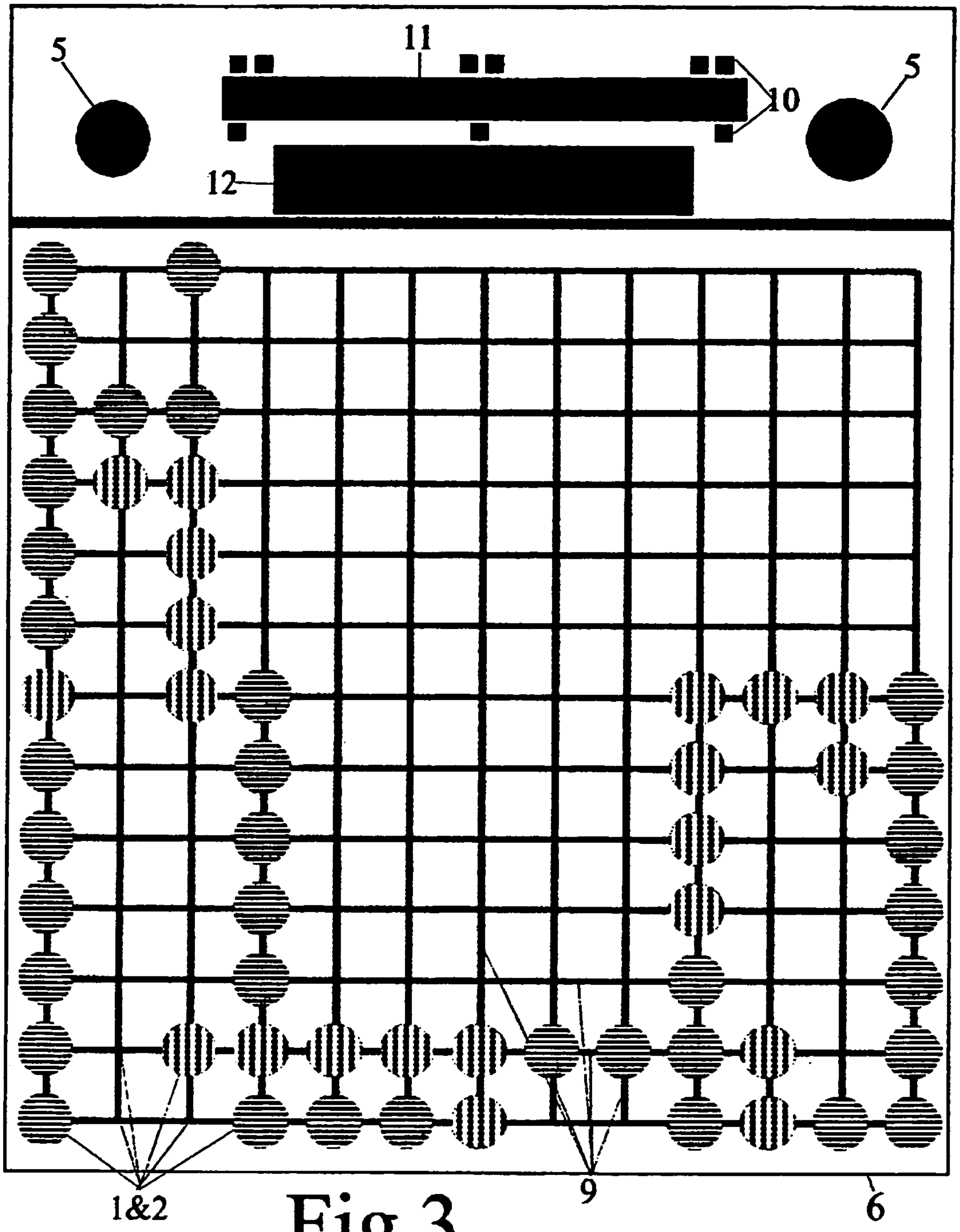


Fig 3

Grid Board For Games

This invention relates to traditional board games like GO, which are played by putting 'stones' (pebbles) on a board, and to new games which are played in the same way. It presents an electronic board to play these games.

Games like GO are played by each player, in his turn, adding a pebble to the board, on one of the points in a grid of lines drawn on the board, or in one of the squares on the board. These games have the advantages of being based on simple rules and being interesting intellectually. Their disadvantages are:

1) They require somewhat tricky movement when putting the stone on the board in the right place without disturbing other stones.
2) They tend to suffer from delays when a player is thinking on a move.
3) Some of the moves require additional 'housekeeping' operations, e.g. taking stones of the board in GO or changing them to stones in another colour in other games.
4) The players need to keep the rules and do the counting of stones themselves, which puts extra demand on the players.
5) The stones are separate objects, which are easily lost.

Disadvantages 3-5 can be solved by programming a computer to display the board and stones. The program would be simple enough that it can be put on a small and cheap CPU, and hence be built into a standalone playing board. In principle, the computer could also limit the time allocated to each player, thus solving disadvantage 2.

The problem of input (disadvantage 1) , however, is not solved so well by current electronic systems. That is because input for existing electronic systems is done through buttons, or other devices, which are separated from the display. For games where there is a small repertoire of possible different inputs this is acceptable, but for board games like GO, there are many possible different inputs (the number of points in the grid). Inputting a point on buttons off the display requires the players to perform some mental operation to convert the point they think about to the right input. This is relatively slow and error-prone process. For slow-going games that is annoying but acceptable, but it makes it impossible to play fast board games on these systems. The aim of the *grid board* is to solve this problem, by making the display of the state of the game (the board) also the place where the input from the players is received.

The conceptual structure of the grid board is sketched in Figure 1. According to the current invention (the *grid board*), the grid board is made of *grid points* 1 & 2 which are mounted in a grid on a flat surface 6. Each grid point is a soft button or a sensor 1, which can detect when it is touched, and can be lit by two or more different lights 2. The sketch shows only 3 grid points for clarity, but the actual board has many more grid points (typically 49 - 1000). All the grid points are connected to a *controller* 3, which is a small CPU + memory. When a grid point is touched, the controller is notified (arrows from buttons 1 to the controller 3), and the controller controls which lights are on (arrows from the controller 3 to the lights 2). The controller is programmed to manage various games, puzzles and exercises. Managing a game means that the board displays the state of the game by putting on the appropriate lights to mark which points are 'occupied by stones'. When a point is touched, the controller interprets it as 'putting a stone' in this point. It computes the implication according the rules of the game, and changes the lights to reflect the new state of the game.

To allow the players to change the game, change the rules and change other parameters (like time of a turn), the board will need a control area 4, which contains input buttons and character and digit displays. The controller receives information from the control area about which input buttons were pressed, and controls what is displayed.

The grid board will also need a way to signal whose turn it is, which would typically be done by two *turn lights* 5, which are in two separate colours, corresponding to two of the colours of the lights in the grid points. The controller controls these turn lights, and signal to the players whose turn it is by lighting the corresponding turn light.

The board will be programmed to play various different games, with adjustable rules and parameters. These include traditional games like GO, with the addition that the time allocated to each move can be limited, and that the board can be set to 'salt' the game by doing random changes to the board. In addition the board can be programmed to play puzzles, single-player games, two-player games between two human players or between a human and the board, and multi-player games. In particular, the grid board can be used to implement *fluid* board games, where the board changes even when the players do not play (see for example the description of the 'Game of Life' on p. 4). Currently, there is no realistic way to implement these kind of games except when the number of possible different inputs is small.

The arrangement of grid points would be in most cases square, but can also be of different shapes (e.g. rectangular, hexagonal, triangular). The overall shape of the board can also vary. In mass production, the buttons themselves would probably be made of a transparent membrane keyboard, mounted on a PCB board on which two LEDs per button are mounted as well.

A specific embodiment of the invention will now be described with reference to the accompanying drawings:

Figure 1 shows the conceptual structure of the board.
Figure 2 shows a sketch of the electronic components of an example board.
Figure 3 is a sketch of the way the board looks for players from above.

All the components of the example board, which are shown in Figure 2, are standard components. All the user interaction (grid points 1& 2, control area 4 and turn lights 5) is implemented by mounting a transparent touch screen 7 on top of a LCD screen 8. The two screens together correspond to the surface 6 in Figure 1, and comprise the top of a flat box, which contains the controller 3 (A single board computer, including CPU, RAM and solid state disk) and the power supply. Both of the screens are connected to the controller 3. The software uses standard graphic routines to display all the board user interface (shown in figure 3) via the LCD screen 7, and interprets the coordinates of input from the touch screen 8 as pressing buttons. A 'light' (representing a 'stone') in a grid point is shown by drawing a circle of the right colour in the grid point.

Figure 3 shows a sketch of the board from above 'in a middle of a game', with some grid points occupied. The main area is the grid of 13x13 lines 9 which are drawn in black, and do not change throughout the game. The intersections of the lines are the grid points 1&2. Some of these are 'lit' in one of two colours, i.e. a circle of the appropriate colour is drawn on the intersection (indicated in the figure by two different shading). When a player touches the touch screen near one of the intersections, the software interprets it as 'putting a stone' on this intersection. At the top are the two turn lights 5 and the control area (corresponding to 4 in Figure 1), which contains several input buttons 10, a display of characters 11 to and a display of digits 12.

The electronic design which is described in figure 2 allows a very flexible layout of the user interface, but contains expansive components, the LCD screen and the touch screen, and requires a moderately powerful CPU to perform the graphic operations. Therefore, the user interface is designed such that it is easy to replace the expansive components by simple ones for mass production. The character and digits displays 11 & 12 and the input buttons 10 will be replaced by standard components. The touch screen 7 will be replaced a membrane keyboard, with a button 1 for each grid point. The lighting of each grid point 2 will be done by two LEDs of different colours, and similarly for the turn lights 5. These modifications will eliminate the need for the screens, and the software will not need to perform graphic operations, so a smaller CPU can be used.

The software:

The central loop of the software repeats these four steps:

1) Check if any of the input buttons was pressed. If any button was pressed, perform the appropriate operation (set a parameter, stop the game, start the game).

2) Check if any of the grid points was touched. If so, compute the implications according to the rules of the game, perform all the changes to the board, and then switch the turn to the other player. The computations for the game GO are given below as an example. Switching the turn means switching the turn light of the current player off, setting the internal variable current_player to the other player, switching the turn light of the other player on and setting a variable, the *turn end mark*, to the current time plus the turn time.

3) Check the clock and compare it to various time marks. A time mark is a variable set to some value, which is compared to the current time. The most important is the turn end mark, and if this is passed, switch the turn as in 2. Other time marks are for updates of the displays.

4) Check if there are game specific operations to perform. For example, in the 'Game of Life' (p. 4) this is used to perform generation change.

Operations when a grid point is pressed for the game GO

a. Switch of the turn light of the current player and set a short (Current time + ~200 ms) delay mark.

b. If checking for KO is switched on, check if the move breaks the KO rule. If it does, reject it, which means notify the players via the character display, switch on the turn light of the current player on and return to the main loop.

c. Check if as the result of this move any of the 'stones' of the opponent are captured, i.e. completely surrounded by 'stones' of the current player. If so, 'remove' these 'stones', i.e. switch their lights off.

d. Check if the new 'stone' is part of a group of 'stones' of the current player which is completely surrounded by 'stones' of the other player. If so, reject the move as in (b) above.

e. 'Put on the stone', i.e. switch on the light of the current player in the grid point.

f. Check if the game is decided, i.e. if it is clear which part of the board is controlled by which player. If it is, end the game.

g. Wait until the delay mark which was set in step (a) passes.

h. Update the digit display to show the new count of stones for each player.

i. Pass the turn to the other player, i.e. switch the turn light of the other player on and set the time mark (current time + the time for a move) for this player.

Games implemented in the example of the Grid Board

1) GO.

2) Othello. In this game, when a stone is put, all the stones of the other colour which are between the new stone and another stone of the same colour (without spaces) along a straight line are changed to the colour of the new stone. The winner is the player with the most points on the board in the end.

3) 'Game of Life'. This is a fluid game, i.e. the state of the board changes even if the players do not play. The game proceeds in generations. In each generation, the board checks for each point how many of the eight points around it are 'alive' (switched on), and accordingly decides if the point is going to be alive in the next generation. Thus the pattern of lights in the grid points changes each generation. This is implemented by setting a time mark for a generation period and the game specific check (step 4 in the central loop) performs a generation step when the mark passes. In parallel, the player(s) can bring to life or kill points by touching them.

The 'Game of Life' can be played in one person mode, in which case the player tries to keep the board alive as long as possible, or to kill it as fast as possible, or in two players mode, in which case each player tries to take control of the board.

CLAIMS

1 A grid board, made of many *grid points* arranged in a grid on a flat surface, and connected to a *controller*, which is a small CPU + memory. Each grid point is a soft button or a sensor, which can detect when it is touched, and can be lit by two or more different lights. When a grid point is touched, the controller is notified, and the controller controls which lights are on. The controller is programmed to manage various games, puzzles and exercises. Managing a game means that the board displays the state of the game by putting on the appropriate lights to mark which points are 'occupied by stones'. When a point is touched, the controller interprets it as 'putting a stone' in this point. It computes the implications according the rules of the game, and changes the lights to reflect the new state of the game.

2 A grid board as claimed in Claim 1 with a control area, containing input devices and character and digit display, to allow the user to set which game is played, the exact rules and other parameters, and to display these parameters and the current result (number of 'stones' on the board for each player).

3 A grid board as claimed in Claim 1 or Claim 2, with the addition of two turn lights, which are in two separate colours, corresponding to two of the colours of the lights in the grid points. The controller controls these turn lights, and signal to the players whose turn it is by lighting the corresponding turn light.

4 A grid board as claimed in any preceding claim and programmed to manage GO and/or other traditional board games which are traditionally played by each player adding a stone to the board.

5 A grid board as claimed in any preceding claim and programmed to limit the time that a player can make a move. When the time passes, the controller passes the turn to the other player.

6 A grid board as claimed in any preceding claim and programmed to manage games where the board changes even when the players do not make a move.

7 A grid board substantially as described here with reference to figures 1-3.

Application No:	GB 9907163.1	**Examiner:**	Matt Jefferson
Claims searched:	1 to 7	**Date of search:**	19 July 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.Q): A6H (HLM).

Int Cl (Ed.6): A63F 3/00, 3/02.

Other: Online: EPODOC, PAJ, WPI.

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X Y	GB 2147817	(STC PLC) See whole document.	X:1 & 5. Y: 4.
X	WO 97/26057	(TIGER ELECTRONICS, INC.) See whole document.	1, 3 & 5.
X	US 4279421	(TEPOORTEN ET AL.) See columns 2 to 4 and figures 1 to 4.	1.
Y	US 4244635	(SASAKI ET AL.) See whole document.	4.
Y	JP 04 367684	(TAKANASHI ET AL.) See abstract.	4.

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.